General Purpose SATCOM on the Move Two-Axis Gimbal System for Low Altitude Applications

Progress Report 1 15.04.22

Buğra Coşkun - 180223028
Akanji Leonel - 170223087
Evliya Arslan - 140224016

Kocaeli University, Dep. of Engineering,
Mechatronics Engineering

Note From Buğra Coşkun

This project is heavily inspired from my current project and past experiences I had the privilege of acquiring as a part of Profen Communication Technologies R&D Center working as a control engineer. As such the technical details and design similarities must be first confirmed with the appropriate authorities in the company before I share them in these reports and open source platform. Thus there might be details that are left off in the progress and project reports but me and my group will work hard to fill in the gaps.

1. Introduction

In our project we aspire to create an open source hardware satcom on the move system for a wide range of applications. As the frequency and bandwidth of satellite communication increases the radiation pattern shrinks to sub degree orientations which in turn necessitates the employment of robotic pedestals that can track the satellite and reject outside disturbances. There has not been many research to practical design of such systems and most of the research that goes into such systems are kept behind few company walls. As such we work to create a practical gimbal system that can support a relatively medium sized antenna from low-cost and easily accessible parts.

2. Purpose

The end-goal of this project is the creation of an open source hardware system that can track GEO (geostationary), LEO (low earth orbit) and MEO (medium earth orbit) satellites using their TLE (two line keplerian elements set) by means of cancelling out orientation changes of the host vehicle while simultaneously updating azimuth and elevation coordinates of the satellite and tracking it. For our current goal as a part of this research project we aspire to complete the prototype that can support a 30cm parabolic antenna and orient it using two axis mounted in spherical coordinate orientations while cancelling out the orientation disturbances under the pedestal i.e. vehicle movement.

3. General Contributions

The hopeful future contributions of this project is the widescale distribution of an open source hardware SOTM (Satcom on the move) construction guide, tracking algorithm, parts data and all other relevant information for the creation of such system to help organizations, companies and individuals to meet their needs whatever they may be.

4. Work Plan

[1]Tracking and disturbance rejection algorithm

[2]Material selection and 3D Cad model

[3]Electronic schematics and component selection

[4]Writing the control program

[5]Separate hardware tests

[6]Prototype integration

[7]Hardware in the loop test

[8]Performance analysis

[9]Finalization

5. Team

Buğra Coşkun,
Project lead,
Control algorithm,
Satcom and RF systems,
Prototype integration
Performance analysis

Akanji Leonel
Embedded programming,
Interface design,
Electronics design

Evliya Arslan
Mechanical design,
Materials selection,
Prototype QA and documentation

6. Material Selection

Material for construction: Pure PLA for high rigidity

Material for high temperature areas: Aliminium or ABS

Motor selection: Nema 23 Stepper Motors

Motor drivers: TB6600

Power transfer component: Timing belts

Control unit: Raspberry Pi Pico with Micropython Firmware

Orientation sensor: Adafruit IMU (Exact model will be decided depending stocks)

Antenna: Mesh parabolic antenna with X band LNB

7. Prototype and Experiments

These experiments and simulations are copied over from my (BC) paper for Velocity Control Method for Two-Axis Sotm Antenna.

7.1. Algorithm Test

For assuring ourselves that the calculations we have performed are free of errors we need to simulate a proper scenario. To achieve this without relying on any proprietary software and/or toolboxes we need to first numerically define the basic limits of our system such as the maximum angular acceleration and angular rate allowed. Luckily by just defining the angular accelerations and deriving to get the angular rates about the principal axes of the vehicle we are able to do just that. For this application we produce angular velocities that are 120 degrees apart in phase and sinusoidal.

$$\alpha \doteq \max(\alpha)\cos(at+b) \tag{1}$$

$$\omega = \int \alpha \, dt = \max(\alpha) \int \cos(at+b)\, dt \tag{2}$$

$$= \max(\alpha)\left[\frac{1}{a}\sin(at+b) + C\right] \tag{3}$$

$$\therefore \max(\omega) = \frac{\max(\alpha)}{a} + C\max(\alpha) \tag{4}$$

$$C \doteq 0 \quad \Rightarrow \quad a\max(\omega) = \max(\alpha) \tag{5}$$

As this is just a test to see if the basic algorithm is accurate we can be satisfied with disturbance of equal amplitude and $2\pi/3$ radians apart in phase. A small side note: coordinate axes can be represented by enumeration them as 1,2,3 for x,y,z respectively.

$$\alpha_i \doteq \max(\alpha_i)\cos\left[a_i t + \frac{2\pi}{3}(i-1)\right] \tag{6}$$

$$\therefore\ \omega_{v,i} = \left(\frac{\max(\alpha_i)}{a_i}\right)\sin\left[a_i t + \frac{2\pi}{3}(i-1)\right] \tag{7}$$

So we can use the angular rates to calculate the propagated orientation matrix according to the convention used by XSENS[x].

$$R_v = R_z(\psi)R_y(\theta)R_x(\phi)\ \Rightarrow \tag{8}$$

$$R'_v \doteq R_z(\psi + \omega_{v,z}dt)R_y(\theta + \omega_{v,y}dt)R_x(\phi + \omega_{v,x}dt) \tag{9}$$

Using the propagated orientation matrix and the angular rates we have used to propagate that said matrix we follow the algorithm we have constructed before to calculate the angular rate we want the motors to have. Because that we are working with an ideal system right now we will assume that the motor rates will be whatever we define them to be.

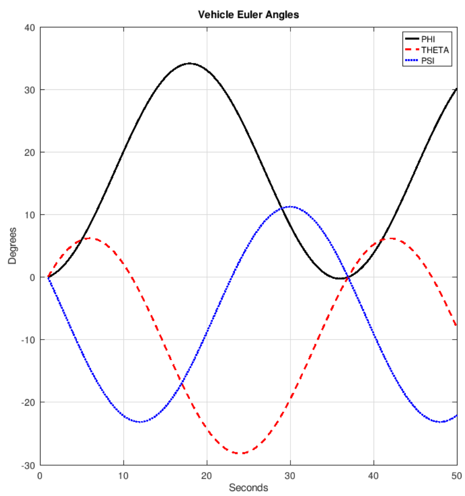

Figure 1: Vehicle Euler Angles

As such we add the true angular rate the LOSv vector is doing about the axes of the vehicle frame and the reference velocity we produce using a PID approach from the position error. Vehicle euler angles in the simulation is given in figure 4 while LOSl vector and manipulator angles are given in figure 5 and the position error is given in figure 6.

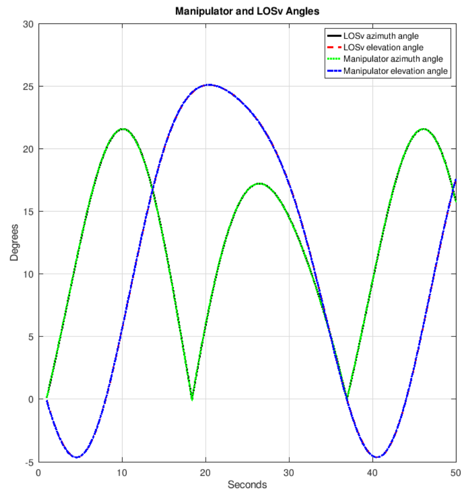

Figure 2: Tracking Results

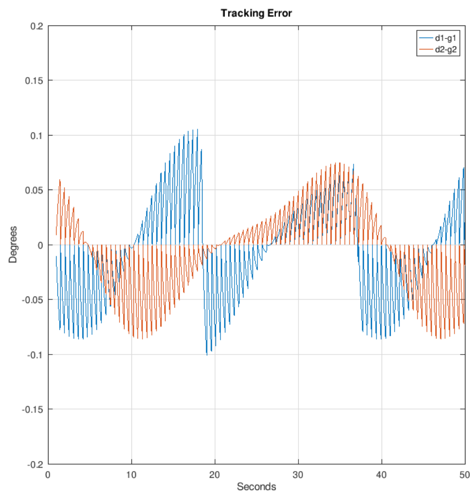

Figure 3: Tracking Error